# A Scalable QoS-based Routing Mechanism for Supporting Multimedia Applications


Luís Henrique Maciel Kosmalski Costa[1,2]
Luis.Costa@lip6.fr

[1] Laboratoire d'Informatique de Paris VI - LIP6
Université Pierre et Marie Curie
4, place Jussieu
75252 Paris Cedex 05
France

Otto Carlos Muniz Bandeira Duarte[2]
otto@gta.ufrj.br

[2] Grupo de Teleinformática e Automação – GTA
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica
P. O. Box 68504 - CEP 21945-970
Rio de Janeiro - RJ - Brasil *



## Abstract

*This paper proposes a scalable algorithm for Quality of Service (QoS) based routing. This type of routing is one of the main components for supporting new multimedia applications in the Internet. The algorithm takes into account the available bandwidth, the transmission delay and the packet loss probability for each link in route computation. This three-metric algorithm is as complex as a two-metric algorithm and provides a better characterization of application requirements. The mechanism was implemented using a network simulator and the results obtained are 25% better than the classical approach.*


## 1 Introduction

The Internet is being used more and more for the transport of multimedia information. Nevertheless, the IP protocol has not been conceived for this kind of utilization. It delivers packets using the best-effort discipline, without any QoS (Quality of Service) guarantee. In order to offer QoS, the IETF (Internet Engineering Task Force) is working on the definition of a new Integrated Services architecture [1, 2]. One important area is the definition of a routing architecture that considers the QoS requirements of applications [3]. In QoS-based routing, the route for each flow considers its traffic characteristics.

A main issue in QoS routing concerns scalability. The scheme must have a complexity that does not grow indefinitely with the increasing number of nodes. Another point is the size of routing tables, important issue even for traditional routing schemes. The definition of routing algorithms considering multiple metrics is very important. Some authors consider the computation of routes with more than two metrics impossible due to its complexity [4].

This paper proposes an algorithm solvable in polynomial time that considers link bandwidth, delay and loss probability. The major advantage of the proposed algorithm is its simplicity. The algorithm was implemented in Berkeley's Network Simulator (NS)[5]. A complete routing scheme could be constructed with the routing protocol and an IP-like network layer. The proposed algorithm and an algorithm that considers the three metrics independently are compared and the results show an average processing time 25% better.

This paper is organized as follows. In section 2, the main requirements for QoS-based routing and algorithmic issues are discussed. Section 3 presents the proposal of algorithms with three metrics. The simulation and tests are shown in section 4. Section 5 presents some final considerations.

## 2 Routing algorithms

This section presents the main requirements for a QoS-based routing architecture and discusses the different ways of constructing QoS routing algorithms for unicast flows.

### 2.1 Routing protocol requirements

There are several requirements for a QoS-based routing architecture [3]. First of all, the routing protocol must support paths with multiple metrics such as delay, bandwidth, jitter, etc. It must also be able to detect and react to changes in QoS characteristics of links and to route even when there are multiple failures in the network.

The routing protocol must be scalable, essential requirement when considering the size of the Internet. The used algorithm must be able to deal with the growing number of nodes and links in the network without


*This work was sponsored by UFRJ, FUJB, PROTEM-CC, CNPq, FAPERJ, CAPES, REENGE and COFECUB

becoming impractical with regard to routers' processing power and memory. The routing tables cannot grow indefinitely.

Other requirements not presented here can be defined, such as global call admission control and load balancing [6, 3].

### 2.2 Single mixed metric

Algorithms for a single-metric route computation are well known. A direct approach would be to map the user's QoS requirements into a single measure, which could be used by traditional algorithms.

The single mixed metric can be defined as a mathematical function that has a set of QoS requirements as parameters and gives as result a single measure. A function that calculates a value proportional to the desired bandwidth and inversely proportional to delay and loss probability can be defined as

$$f(p) = \frac{B(p)}{D(p) \times L(p)}, \quad (1)$$

where $p$ is the computed path, $B(p)$ its bandwidth, $D(p)$ the transmission delay and $L(p)$ the packet loss probability. The path that results in the greater value of $f(p)$ would be the best route.

However the single mixed metric does not guarantee all QoS requirements. Guaranteeing that $f(p)$ is respected does not ensure that each QoS requirement is respected. Another problem is the different composition rules of various QoS parameters. There are three common composition rules: additive, multiplicative and concave. If the metric for a path is the minimum value between all link metrics, the composition is concave. The additive and multiplicative cases are straightforward. Delay transmission has an additive composition rule, but link bandwidth is concave. So the single mixed metric function defined above does not follow any of these composition rules.

### 2.3 Multiple metrics

The use of multiple metrics models the network in a more precise way, but the problem of finding a path given multiple restrictions can be very complex [4, 7, 8]. Metrics like transmission delay, bandwidth and loss probability can be considered dependent or not. Algorithms based on dependent metrics were proposed by Ma and Steenkiste [7].

Another way is to consider independent metrics. In this case, the computational complexity of the route discovery algorithm is the problem. The larger is the QoS requirements set the more complex the routing algorithm. The algorithm complexity depends on the composition rules of the metrics.

The most usual QoS requirements are transmission delay, jitter, cost, packet loss probability and bandwidth consumption. The link bandwidth has a concave composition rule. The composition rule for loss probability is more complex. Let $p$ be a path, $lp(p)$ the path metric and $lp(i, j)$ the metric associated with the link between node $i$ and node $j$. The composition metric for loss probability is shown in equation 2.

$$lp(p) = 1 - ((1 - lp(i, j) \times (1 - lp(j, k)) \times ... \times (1 - lp(l, m))). \quad (2)$$

However transmission success probability has a multiplicative composition rule. Wang and Crowcroft [4] investigated the complexity of algorithms with additive and multiplicative metrics, and have shown that the problem is NP-complete. Any combination of $n$ additive metrics with $m$ multiplicative metrics is also NP-complete. As a result, between delay, jitter, cost, loss probability and bandwidth, the only solvable combination is that of bandwidth with one other metric, since bandwidth has a concave composition rule.

## 3 The SMM approach

In the previous section, the main problems to define a QoS-based routing protocol were presented. The use of a single mixed metric would be discarded, due to the difficulties imposed by the different composition rules intrinsic to each metric.

Another important scalability issue is the size of routing tables. In QoS routing, the route decision is made accordingly to traffic characteristics of flows. A natural approach is the construction of tables with one route per flow. This would lead to large routing tables, with size proportional to the number of nodes and to the amount of users connected to each node.

A tradeoff between the precision of the route used by a flow and the size of the tables is the maintenance of a set of routes to each destination. Each route represents a class of service. QoS requirements of individual flows have to be mapped into one class. The routing table size will be larger than the actual routing tables, but smaller than tables with one entry per flow.

Given the observations made above, an hybrid scheme can be proposed. This mechanism is a tradeoff between the algorithm simplicity and route precision given the user's QoS parameter set. Some changes are made to the algorithm presented in [4] by means of a careful utilization of a single mixed metric.

### 3.1 Metric combination

The principal metrics considered until now are delay, jitter, bandwidth and loss probability. Delay and jitter both have an additive composition rule. Loss probability has a more complex composition rule, as shown in section 2. Transmission success probability is multiplicative, but if the logarithm function is applied, the composition rule turns additive and it is possible to combine this metric with other metrics that are also additive. This eliminates the composition rule problem.

It is important to note the effects of this transformation. The metric value for the loss probability is always between 0 and 1. So when the logarithm is calculated, this interval is transformed into $-\infty$ to 0.

Numbers with a so large absolute value would constitute a problem for route computation. But if the interval for the transmission success probability is limited, say from 0.1 to 1, the logarithmic interval will be bounded from $-1$ to 0. This limitation is reasonable: this means that transmission success probability will not be less than 10%. Therefore it is possible to consider that the absolute value for the logarithmic success probability will be from 0 to 1. The same interval is valid for the path metric (composed of various links). The success probability of a path will be smaller than the smallest metric of each link on the path, and will approximate zero as the number of links increase.

The transmission success probability will be represented by real numbers from 0 to 1, and delay by integers. So it is possible to distinguish the contribution from both metrics at the single mixed metric (the other disadvantage mentioned above).

With a single mixed metric for delay and loss probability, it is possible to construct algorithms that also take into account the bandwidth. The idea is to execute a previous search in the network graph to eliminate all arcs that do not meet the bandwidth restriction. The next step is to use an algorithm similar to Dijkstra's algorithm with some changes that will guarantee also that all combined metrics are individually respected.

### 3.2 The SMM algorithm

In this section $SMM$ (from Single-Mixed-Metric routing algorithm) is presented. The algorithm considers available bandwidth, propagation delay and loss probability in route computation.

Let $G = (N, A)$ be a representative graph for the network, with $N$ nodes and $A$ arcs. Two real numbers are associated with each arc, $b_{ij}$ representing its available bandwidth and $u_{ij}$ (equation 3) being the single mixed metric defined as the addition of $Slog(i,j)$, the logarithmic transmission success probability (real number between 0 and 1) with $D(i,j)$, the propagation delay (an integer).

$$u_{i,j} = Slog(i,j) + D(i,j). \tag{3}$$

When an arc $(i,j)$ is not present in the graph, it is considered that $b_{ij} = 0$ and $u_{ij} = \infty$. Given a path $p = (i,j,k,...,l,m)$, the path width, $w(c)$, is given by $min[b_{ij}, b_{jk}, ..., b_{lm}]$ and its length is the addition of the single mixed metric related to each of its component links, $l(p) = u_{ij} + u_{jk} + ... + u_{lm}$. Given two constants $B$ for bandwidth and $U$ for the maximum acceptable value for the single mixed metric, the problem is reduced to find a path $p$ between $i$ and $m$ such that $w(c) \geq B$ and $l(c) \leq U$.

Step 1: $\forall i,j : u_{ij} = \infty$ if $b_{ij} = 0$.
Step 2: $L = \{1\}; \forall i, i \neq 1 : U_i = u_{1i}$.
Step 3: Find $k \notin L \mid U_k = \min_{i \notin L} U_i$.
 If $U_k > U$, a path cannot be found: end.
 If $m \in L$, a path was found: end.
 $L := L \cup \{k\}$.
Step 4: $\forall i \notin L : U_i := \min[U_i, U_k + u_{ki}]$.
 Go to step 3.

This algorithm does not ensure that the transmission success probability is larger than the desired minimum (or smaller, in log). However, since delay is an integer, is guaranteed. But if there are two paths with the same delay, than it is guaranteed that transmission success probability will be maximum.

The loss probability QoS requirement is mapped into a maximum absolute value for the logarithmic transmission success probability. In other words, the addition of these metrics for each link on the path should be smaller than a known bound.

Let $P$ be the known limit for logarithmic transmission success probability and $Slog(ij)$ the logarithmic transmission success probability for arc $ij$. Given a path $p$ with $x$ hops, if $Slog(ij)$ for each link is smaller than $P/x$, then $Slog(c)$ will be smaller than $P$. However, when a path is being computed, its length ($x$) is not known yet. A first idea is to restrict $Slog(ij)$ to $P/C$, where $C$ is the length of the longest path on the network graph. If $Slog(ij)$ is limited to $P/C$ at each step, then it is guaranteed that $Slog(p)$ will be smaller than $P$. However this limit can be too restrictive. The route between two nodes can have in the general case a length smaller than $C$, so its logarithmic transmission success probability is more restricted than it would be needed.

An alternative scheme can relax this restriction. At each step, a hop "remainder" ($SS$) and a logarithmic probability "remainder" ($SP$) are defined. The hop remainder is the difference between the length of the longest path in the graph and the length of the actual computed path. The probability remainder is the difference between the required limit and the actual value for the logarithmic transmission success of the path at the actual step. Then the maximum logarithmic probability at each step will be $SP/SS$ instead of $P/C$. This value can be larger than $P/C$, in the case where $SP$ is smaller than $SP \times (P/C)$. So the probability of finding a path in the network graph grows. The algorithm modified is:

Step 1: $\forall i, j : u_{ij} = \infty$ if $b_{ij} = 0$. $SS = C; SP = P$.
Step 2: $L = \{1\}; \forall i, i \neq 1 : U_i = u_{1i}$.
Step 3: Find $k \notin L \mid (U_k = \min_{i \notin L} U_i)$ and $((U_k - \lfloor U_k \rfloor) < (SP/SS))$.
If $U_k > U$, a path cannot be found: end.
If $m \in L$, a path was found: end.
$L := L \cup \{k\}$.
$SS := SS - 1$. $SP = SP + (U_k - \lfloor U_k \rfloor)$.
Step 4: $\forall i \notin L : U_i := \min[U_i, U_k + u_{ki}]$.
Go to step 3.

This way, the transmission success probability restriction is still stronger than the real QoS requirement, but all QoS requirements are guaranteed without increasing of complexity. This imprecision in route discovery is a tradeoff with the algorithm's complexity.

The $SMM$ algorithm takes the same number of iterations to find a route as the original algorithm for two metrics [4]. The computational complexity added by the single mixed metric calculation is minimum, since this value can be computed before the iterative algorithm. The scalability for the proposed routing approach is achieved by the use of the $SMM$ algorithm in addition with class of service based routing tables.

## 4 Implementation in NS

The proposed scheme was implemented in NS [5]. The distance-vector unicast protocol was used as the basis for the implementation since it supports link dynamics.

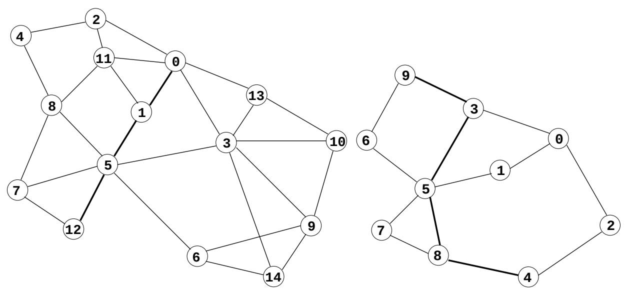


Figure 1: Test topologies.

The QoS-based routing scheme was implemented in the NS simulator by modifying its code. First, link implementation was modified to admit three metrics: propagation delay, available bandwidth and packet loss probability.

A packet classifier that takes into account the type of packet was created. Each class of service was defined as having one metric as the most important: delay, bandwidth or loss probability. The routing table now has three entries per destination.

Two approaches were implemented. In the first one the route is searched by checking the metrics one by one. This approach is referenced as $3M$. The second approach uses the $SMM$ algorithm. The routes found are not necessarily the same to all source and destination pairs, but they always comprise the QoS requirements defined for each class of service.

The NS routing mechanism does not have real-time support, so some changes had to be made. The idea is to measure the time spent to find routes, that depends on the number of iterations performed by the algorithm. A delay proportional to the number of steps spent by the algorithm was introduced before route updates.

Two topologies (shown in figure 1) were constructed each of them with one traffic source that sends packets at a constant bit rate (CBR). A link goes down at a determinate moment. This link is one of the component arcs of the route followed by the traffic. After that, the routing protocol has to recompute the routes. While the routes are not updated, the source continues to send traffic on a failed route. A quantity of packets proportional to the time the routes are not updated will be lost (the traffic generated is CBR).

In the fifteen node topology the CBR traffic source sends packets from node 0 to node 12. The route followed by the packets is $0-1-5-12$. The percentage of lost packets when using the two approaches and the percentile difference between them were measured. As shown in table 1, the results obtained by the $SMM$ algorithm are better than the $3M$ approach.

Table 1: Packet losses for the fifteen node topology.

| | Algorithm | | Percentile Difference |
|---|---|---|---|
| Failure | $3M$ | $SMM$ | % |
| Link 0-1 | 13.1 | 9.0 | 31 |
| Link 1-5 | 15.3 | 11.8 | 24 |
| Link 5-12 | 14.9 | 8.9 | 40 |

Other experiments were made with the smaller topology with ten nodes. The source sends packets from node 9 to node 4, following the route $9-3-5-8-4$. In this case, two different tests were made. In the first, the set of metrics assigned to the links in the topology are very similar, in the other the metrics are very different among them. These tests aimed to show the influence of the set of metrics in the algorithm performance.

It is expected that the exhaustive search used by the $3M$ approach will have a poor performance when the metrics are very similar. When it happens that two possible paths have the same metric in that scheme, the decision is made upon another metric. This problem is minimized in the $SMM$ algorithm. Since it uses a single mixed metric, the route decisions take a smaller quantity of steps. The results obtained by the two algorithms are shown in table 2.

Table 2: Packet losses for the ten node topology.

| | Algorithm | | Percentile Difference |
|---|---|---|---|
| Failure | $3M$ | $SMM$ | % |
| Different metrics | | | |
| Link 9-3 | 10.1 | 8.6 | 15 |
| Link 3-5 | 2.2 | 2.2 | 0 |
| Link 5-8 | 4.1 | 4.1 | 0 |
| Link 8-4 | 9.4 | 6.4 | 32 |
| Similar metrics | | | |
| Link 9-6 | 7.9 | 4.9 | 38 |
| Link 6-5 | 11.1 | 8.1 | 27 |
| Link 5-8 | 12.0 | 7.1 | 41 |
| Link 8-4 | 16.1 | 11.2 | 30 |

In the case where the metrics are similar the advantage experimented by the $SMM$ approach is larger. It is important to note that the $SMM$ algorithm performs as good as the $3M$ approach in the worst case.

The advantage of the $SMM$ algorithm enlarges when the metrics are more similar between them, and when the topology size grows. This can be evaluated by the comparison with the results for the fifteen node topology. The average advantage is 25% for the $SMM$ algorithm versus the $3M$ approach.

## 5 Final considerations

QoS-based routing must be considered as a main component to provide new service classes on the Internet. This routing scheme has several requirements and each of its components has specific questions to be considered. Protocol scalability is a major issue of such a routing mechanism, and it is direct related to the used route discovery algorithm.

In this work, after studying the solutions proposed in the literature, an algorithm that takes into account three metrics on route computation, independently, and with the same complexity as two metrics algorithm was constructed. This is done with simple mathematical calculations, which can be performed in advance to route discovery. The utilization of three metrics guarantees that a larger number of applications will be well characterized than with just two metrics.

The simplicity of the proposed scheme is its most important characteristic. Scalable QoS-based routing can only be provided by schemes at the maximum as complex as today's routing mechanisms. The experiments together with the analysis made showed that the $SMM$ algorithm scales very well. Compared to a classical approach, the gain is as large as 25%.

## References


[1] R. Braden, D. Clark, and S. Shenker, *RFC1633: Integrated Services in the Internet Architecture: An Overview.* Network Working Group, June 1994.

[2] P. P. White, "RSVP and integrated services in the internet: A tutorial," *IEEE Communications Magazine*, vol. 35, pp. 100–106, May 1997.

[3] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, *RFC2386: A Framework for QoS-Based Routing.* Gigapacket Networks, Arrowpoint, NEC USA e IBM, Aug. 1998.

[4] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1228–1234, Sept. 1996.

[5] K. Fall and K. Varadhan, *NS notes and documentation.* University of California at Berkeley/Lawrence Berkeley National Laboratory, Sept. 1998. Available at http://www-mash.cs.berkeley.edu/ns.

[6] R. Guerin and A. Orda, "QoS-based routing in networks with innacurate information: Theory and algorithms," in *IEEE INFOCOM'97*, (Kobe), pp. 75–83, Apr. 1997.

[7] Q. Ma and P. Steenkiste, "Quality of service routing for traffic with performance guarantees," in *IFIP Fifth International Workshop on Quality of Service*, (New York), pp. 115–126, May 1997.

[8] R. Vogel, R. G. Herrtwich, W. Kalfa, H. Wittig, and L. C. Wolf, "QoS-based routing of multimedia streams in computer networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1235–1244, Sept. 1996.